CMPG-767 Digital Image Processing

# BM3D FILTERING

# BM3D – Block Matching 3D Filtering

- K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3D transform-domain collaborative filtering,” IEEE Transactions on Image Processing, vol. 16, No. 8, pp. 2080-2095, August 2007
- http://www.cs.tut.fi/~foi/GCF-BM3D/

# BM3D – Block Matching
# 3D Filtering

- An enhanced sparse representation in transform domain.
  - ➢The enhancement of the sparsity is achieved by similar 2-D image fragments(e.g.,blocks) into 3-D data arrays which we call “groups.”

- Collaborative filtering is a special procedure developed to deal with these 3-D groups.
  - ➢Using the three successive steps:
    - 3-D Fourier transformation of a group of blocks
    - Shrinkage of the 3-D Fourier transform of each group
    - Inverse 3-D Fourier transformation

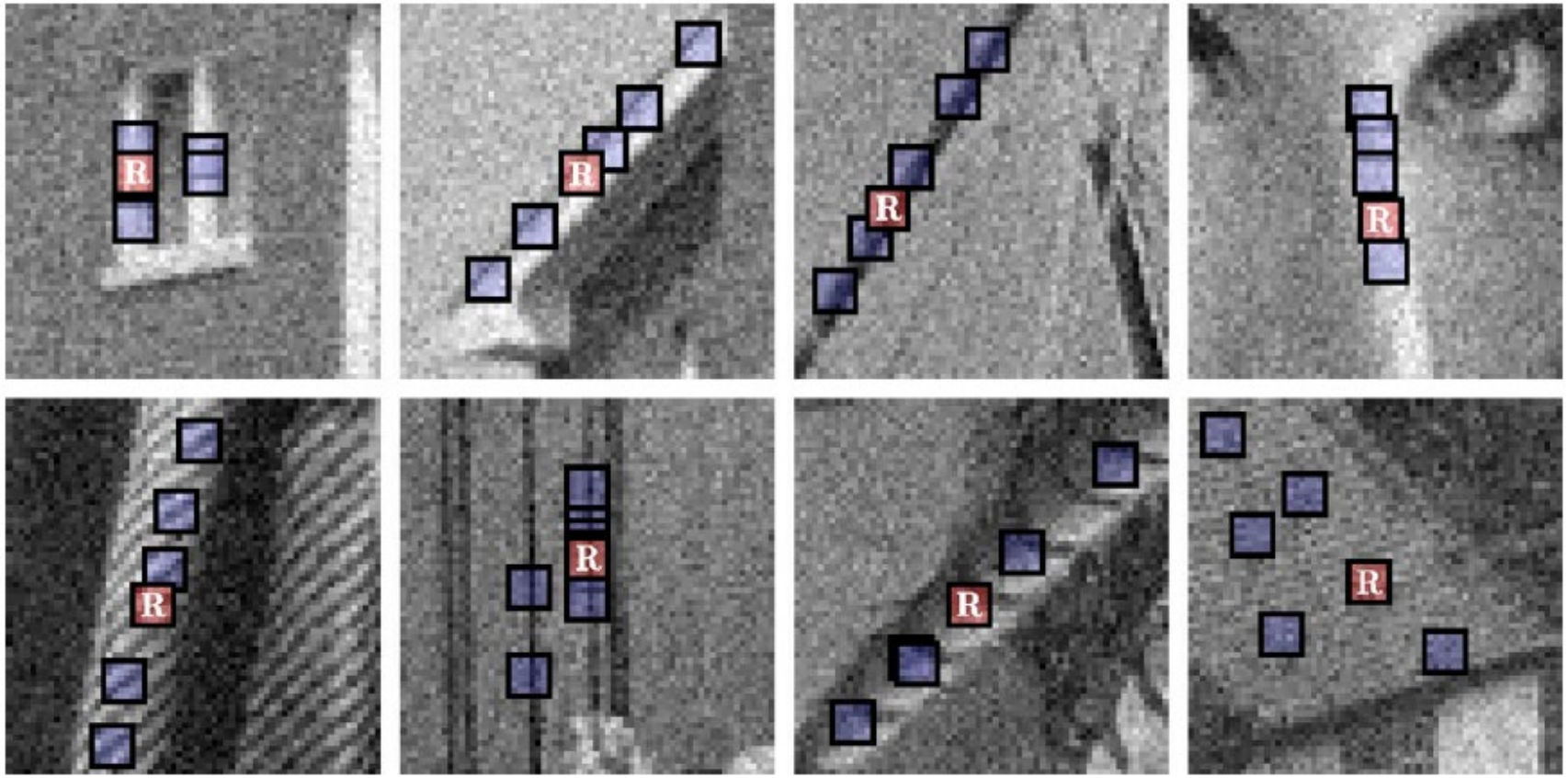
# Grouping

- Grouping is the concept of collecting similar $n$-dimensional fragments of a given signal into an $n$+1-dimentisonal data structure based on some similarity criterion (different criteria can be used)
- The importance of grouping is to enable the use of a higher dimensional filtering of each group , which exploits the potential similarity between grouped fragments in order to estimate the true signal in each of them.
  - This approach is referred to as collaborative filtering

# Grouping by matching

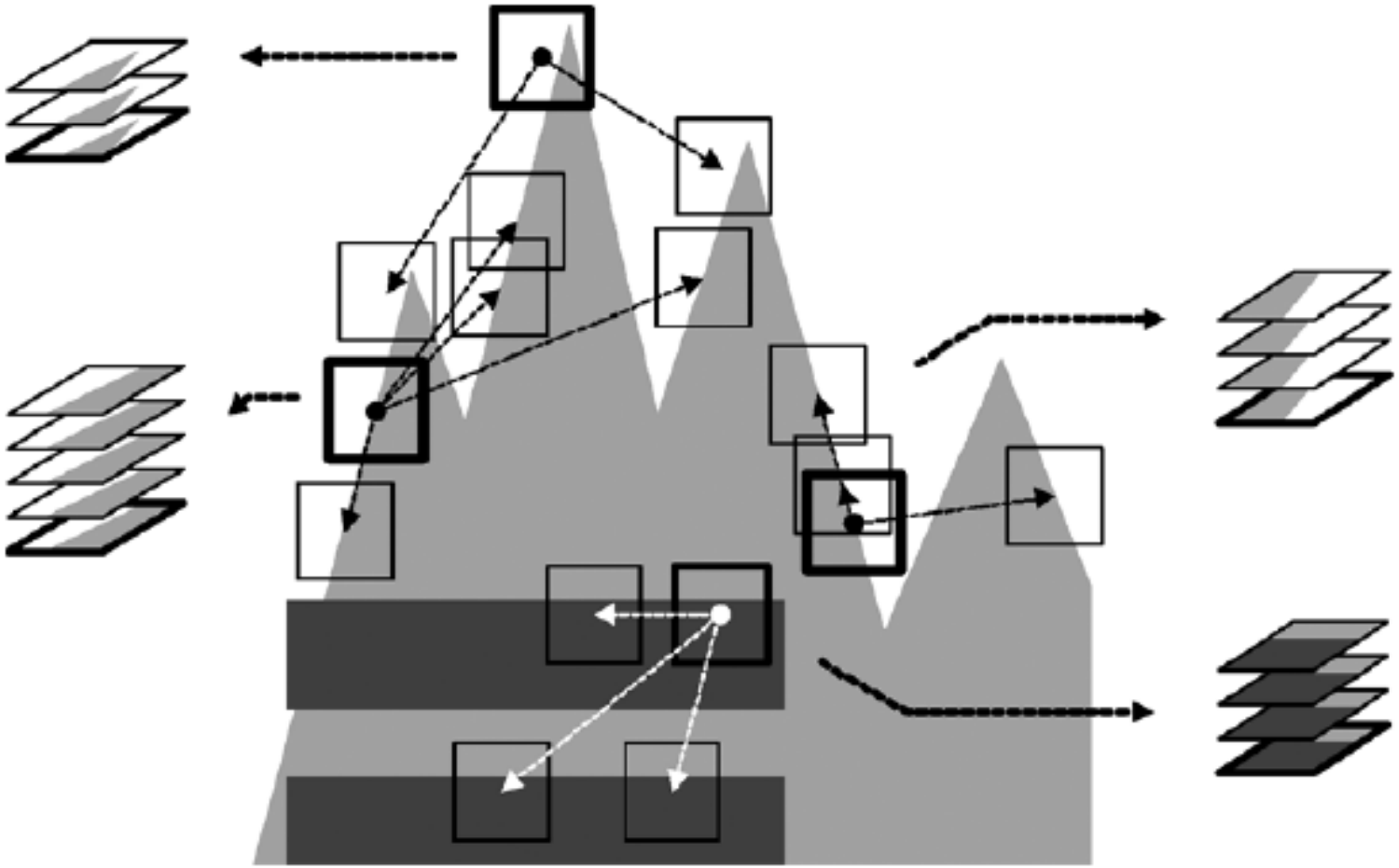
- Matching is a method for finding signal fragments similar to a given reference one
  - Achieving by pairwise testing
  - Distance (for example, if SSD – sum of pixel-wise squared differences $\sum_{x,y \in B} \left( B_1(x,y) - B_2(x,y) \right)^2$ between blocks $B_1$ and $B_2$ is less than a pre-determined threshold
- Block-matching (BM) is a particular matching approach that has been extensively used for motion estimation in video compression (MPEG 1, 2, and 4, and H.26x). As a particular way of grouping, it is used to find similar blocks, which are then stacked together in a 3-D array (i.e., a group).

# Grouping by matching




R is a “targeting” block, other blocks are groped with this block by matching

# Grouping by matching

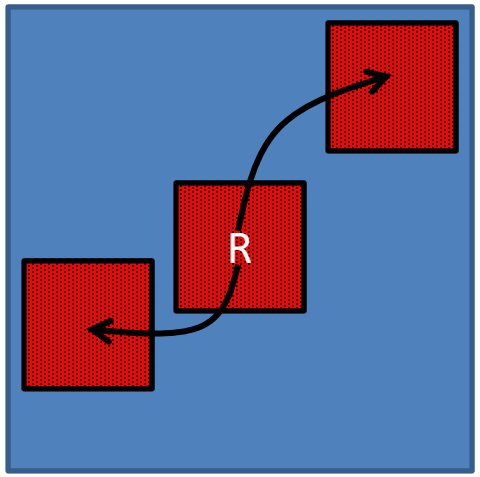


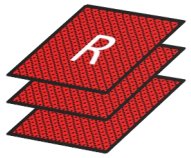
# Collaborative Filtering by Shrinkage in Transform Domain

- Assuming 2D groups of similar signal fragments are already formed, the collaborative shrinkage comprises of the following steps.
  - Apply a 3D Fourier transform to the 3D group
  - Shrink (e.g., by *Wiener filtering*) the transform coefficients to attenuate the noise
  - Apply the inverse 3D Fourier transform to produce estimates of all grouped fragments
- These groups are characterized by both:
  - interfragment correlation which appears between the pixels of each grouped fragment—a peculiarity of natural images
  - interfragment correlation which appears between the corresponding pixels of different fragments—a result of the similarity between grouped fragments

# BM3D Filtering

Block matching




3D grouping

3D transform

Filter / thresholding

Inverse 3D transform

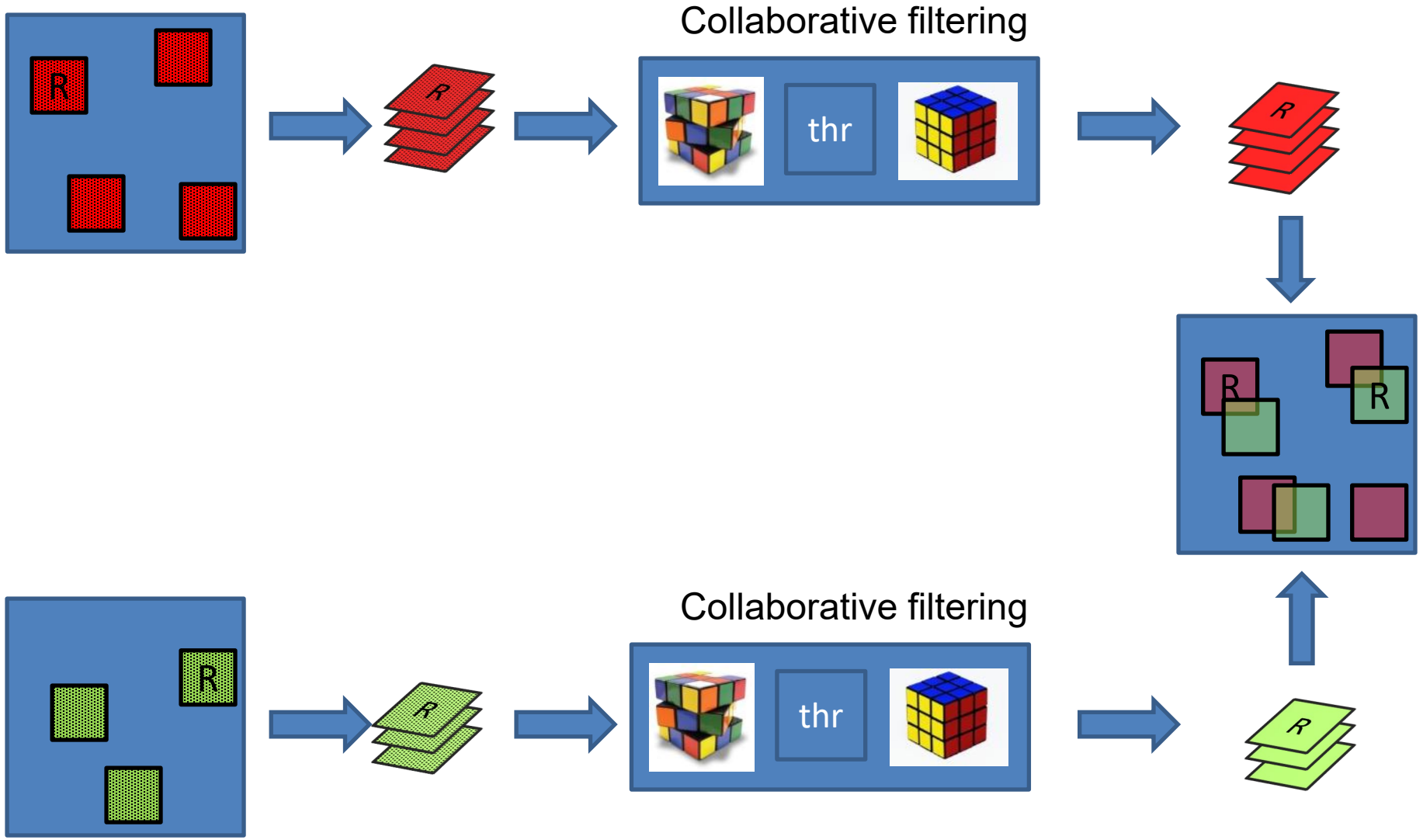

Denoised 3D group

# Collaborative Filtering

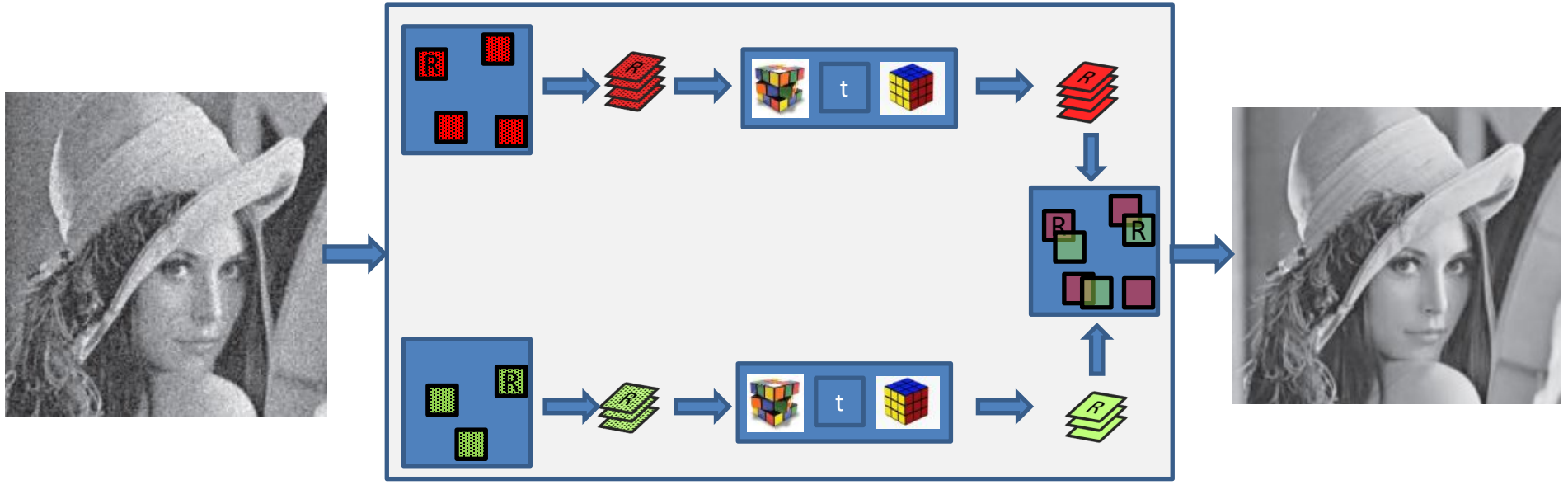
- Use hard thresholding or Wiener filter
- Each patch in the group gets a denoised estimate


- Unlike spatial domain filtering – where only central pixel in reference patch got an estimate

# Multiple BM3D Estimates



# Basic BM3D Filtering



# The benefit of the collaborative shrinkage

- A 2D transform is applied separately to each individual block in a given group of **n fragments**.
- Since these grouped blocks are very similar, for any of them we should get approximately the same number of **significant transform coefficients**.

# Fusion

- Each pixel gets multiple estimates from different groups

- Naive approach
  Average all estimates of each pixel
  …. not all estimates are as good

- Suggestion
  Give higher weight to more reliable estimates

# BM3D - Fusion

- Give each estimate a weight according  to denoising quality of its group
- Quality = Sparsity induced by the denoising

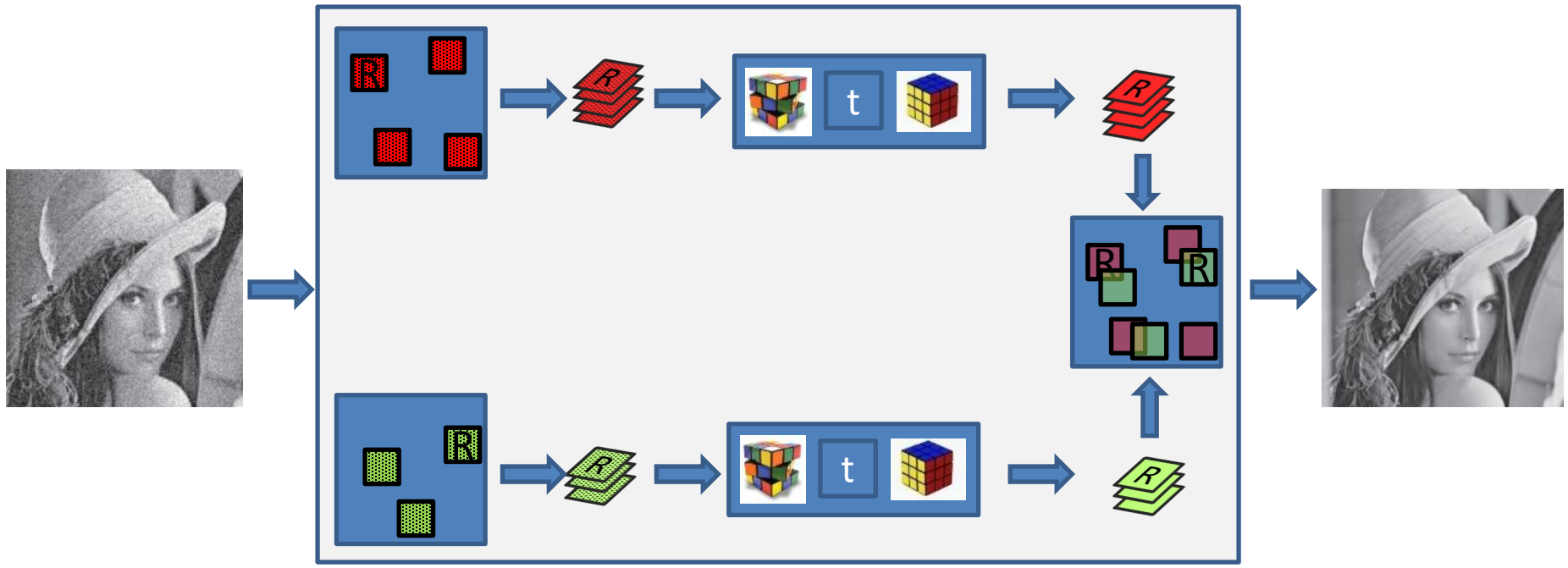
# BM3D in Practice

- Noise may result in poor matching
  ⇒ Degrades de-noising performance

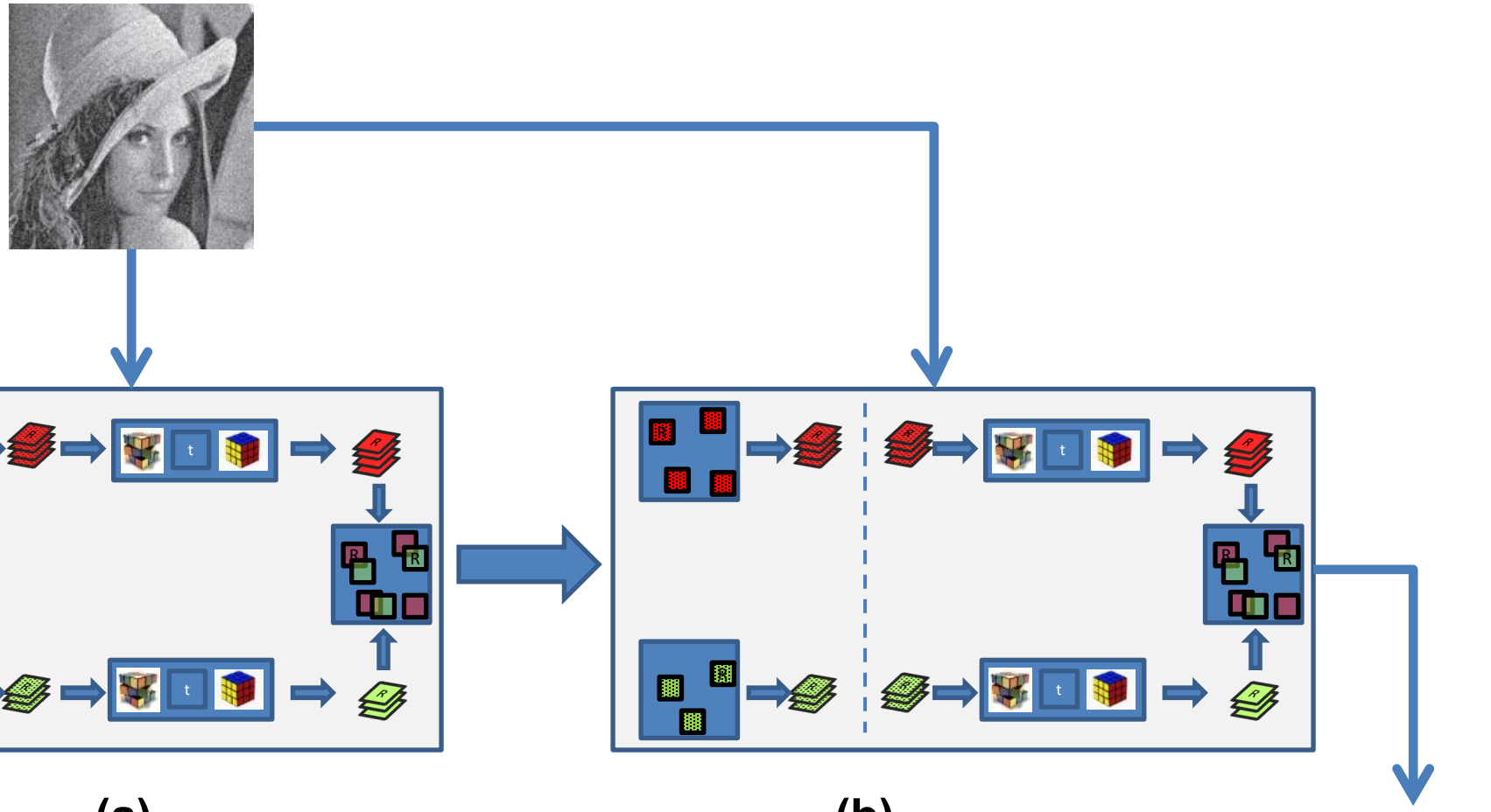
- Improvements:

1. Match using a smoothed version of the image
2. Perform BM3D in 2 phases:
   a. Basic BM3D estimate ⇒ improved 3D groups
   b. Final BM3D

# Basic BM3D Filter



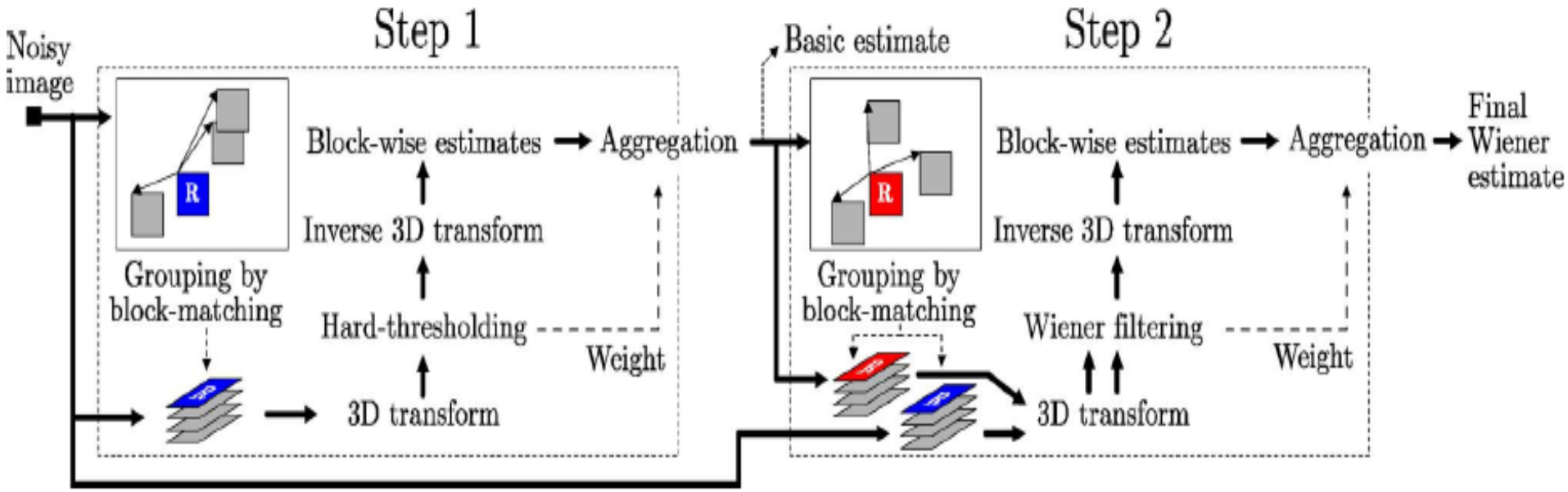
# Two Step BM3D Filtering



**(a)**
Basic denoising:
Hard thresholding

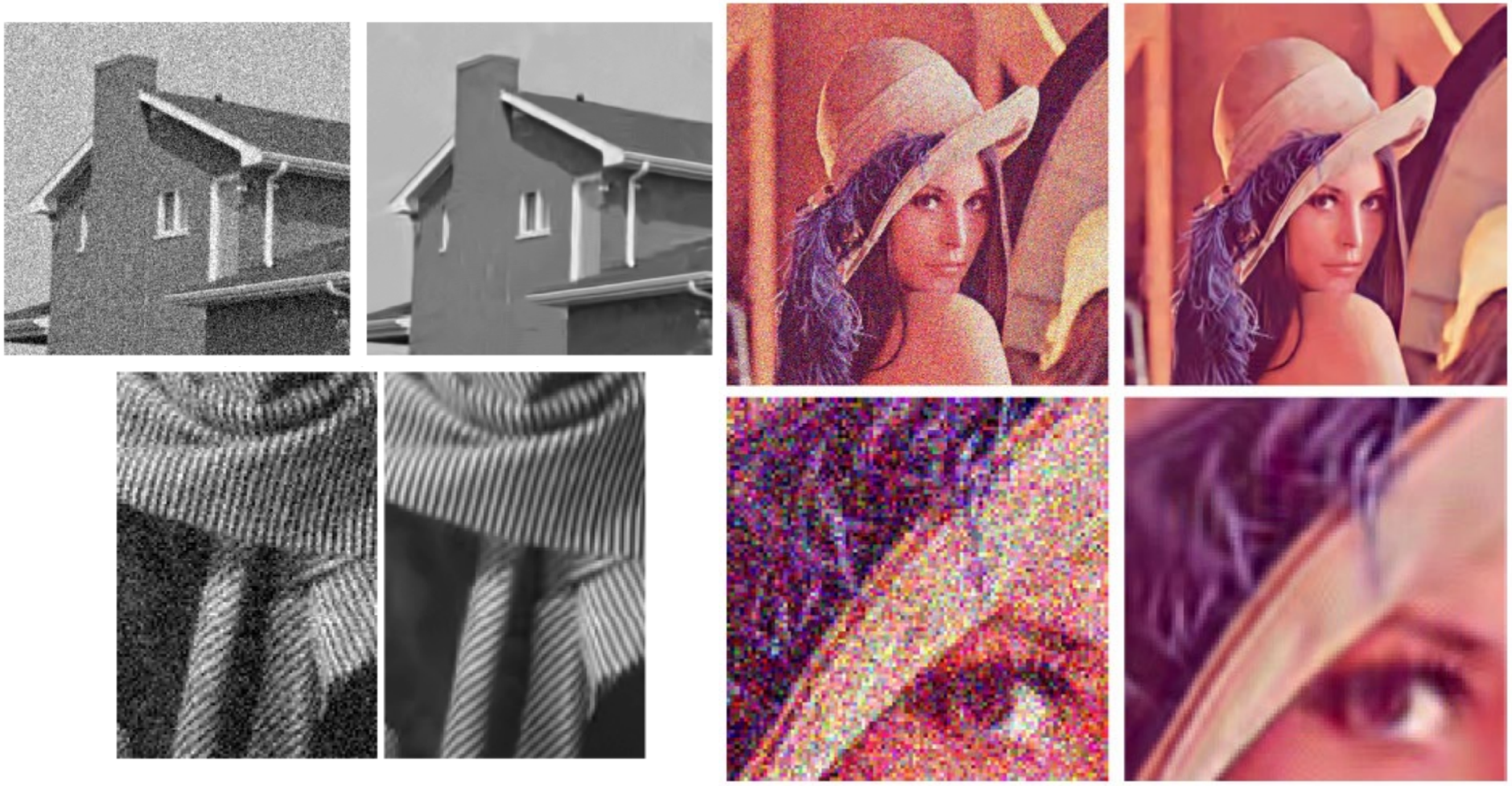
**(b)**
Final denoising:
Wiener filtering

# Algorithm




- The input noisy image is processed by successively extracting reference blocks from it and for each such block:
  - find blocks that are similar to the reference one (**block matching**) and stack them together to form a 3-D array (group)
  - perform **collaborative filtering** of the group and return the obtained 2-D estimates of all grouped blocks to their original locations
- After processing all reference blocks, the obtained block estimates can overlap, and, thus, there are multiple estimates for each pixel. We **aggregate** these estimates to form an estimate of the whole image.

# BM3D Filtering - Examples



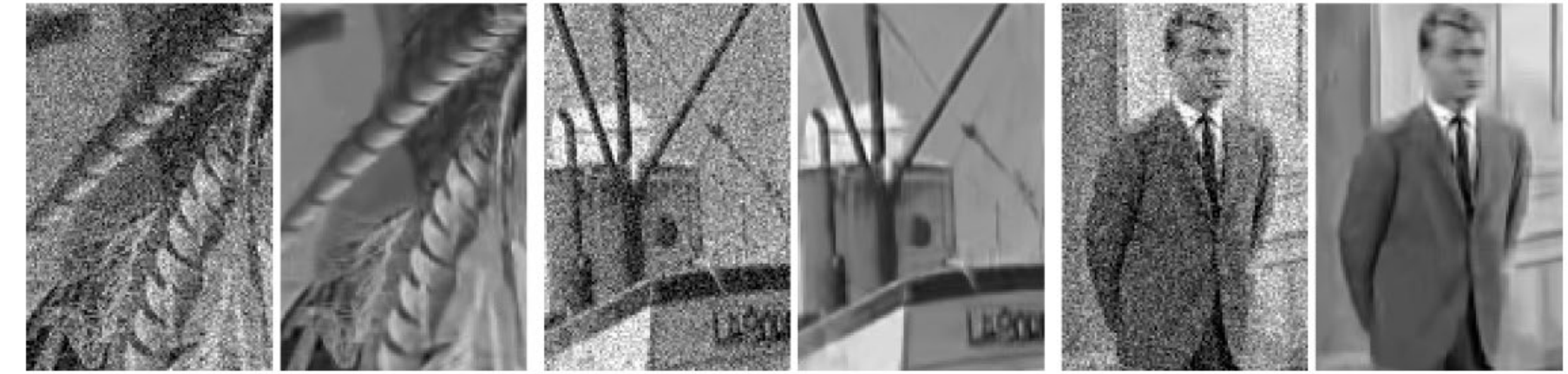
# BM3D Filtering - Examples

Fig. 5. Noisy ($\sigma = 25$) grayscale *House* image and the BM3D estimate (PSNR 32.86 dB).

# BM3D Filtering - Examples

(a) *Lena* (PSNR 32.08 dB) (b) *Barbara* (PSNR 30.73 dB) (c) *Cameraman* (PSNR 29.45 dB)

(d) *Man* (PSNR 29.62 dB) (e) *Boats* (PSNR 29.91 dB) (f) *Couple* (PSNR 29.72 dB)



| Image | $\sigma$ | GF | AF | TV | YNF | EWF | TIHWT | NL-means |
|---|---|---|---|---|---|---|---|---|
| Boat | 8 | 53 | 38 | 39 | 39 | 33 | 28 | 23 |
| Lena | 20 | 120 | 114 | 110 | 129 | 105 | 81 | 68 |
| Barbara | 25 | 220 | 216 | 186 | 176 | 111 | 135 | 72 |
| Baboon | 35 | 507 | 418 | 365 | 381 | 396 | 365 | 292 |
| Wall | 35 | 580 | 660 | 721 | 598 | 325 | 712 | 59 |

29.8dB
29.5dB

# Comparison

| $\sigma$ / PSNR | *Lena* | *Barb* | *Boats* | *Fgrpt* | *House* | *Peprs* |
|---|---|---|---|---|---|---|
| 1 / 48.13 | 48.46 | 48.37 | 48.44 | 48.46 | 48.85 | 48.38 |
| 2 / 42.11 | 43.23 | 43.29 | 42.99 | 43.05 | 44.07 | 43.00 |
| 5 / 34.15 | 38.49 | 37.79 | 36.97 | 36.68 | 38.65 | 37.31 |
| 10 / 28.13 | 35.61 | 34.03 | 33.58 | 32.45 | 35.35 | 33.77 |
| 15 / 24.61 | 33.90 | 31.86 | 31.70 | 30.14 | 33.64 | 31.74 |
| 20 / 22.11 | 32.66 | 30.32 | 30.38 | 28.60 | 32.39 | 30.31 |
| 25 / 20.17 | 31.69 | 29.13 | 29.37 | 27.45 | 31.40 | 29.21 |
| 50 / 14.15 | 28.61 | 25.48 | 26.38 | 24.16 | 28.26 | 25.90 |
| 75 / 10.63 | 26.84 | 23.65 | 24.79 | 22.40 | 26.41 | 24.00 |
| 100 / 8.13 | 25.64 | 22.61 | 23.75 | 21.22 | 25.11 | 22.66 |

| $\sigma$ / PSNR | *C.man* $256^2$ | *House* $256^2$ | *Peppers* $256^2$ | *Montage* $256^2$ | *Lena* $512^2$ | *Barbara* $512^2$ | *Boats* $512^2$ | *F.print* $512^2$ | *Man* $512^2$ | *Couple* $512^2$ | *Hill* $512^2$ | *Lake* $512^2$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 / 42.11 | 43.96 | 44.63 | 43.48 | 46.47 | 43.59 | 43.66 | 43.18 | 42.90 | 43.61 | 43.17 | 43.04 | 43.02 |
| 5 / 34.16 | 38.29 | 39.83 | 38.12 | 41.14 | 38.72 | 38.31 | 37.28 | 36.51 | 37.82 | 37.52 | 37.14 | 36.58 |
| 10 / 28.14 | 34.18 | 36.71 | 34.68 | 37.35 | 35.93 | 34.98 | 33.92 | 32.46 | 33.98 | 34.04 | 33.62 | 32.85 |
| 15 / 24.61 | 31.91 | 34.94 | 32.70 | 35.15 | 34.27 | 33.11 | 32.14 | 30.28 | 31.93 | 32.11 | 31.86 | 31.08 |
| 20 / 22.11 | 30.48 | 33.77 | 31.29 | 33.61 | 33.05 | 31.78 | 30.88 | 28.81 | 30.59 | 30.76 | 30.72 | 29.87 |
| 25 / 20.18 | 29.45 | 32.86 | 30.16 | 32.37 | 32.08 | 30.72 | 29.91 | 27.70 | 29.62 | 29.72 | 29.85 | 28.94 |
| 30 / 18.59 | 28.64 | 32.09 | 29.28 | 31.37 | 31.26 | 29.81 | 29.12 | 26.83 | 28.86 | 28.87 | 29.16 | 28.18 |
| 35 / 17.25 | 27.93 | 31.38 | 28.52 | 30.46 | 30.56 | 28.98 | 28.43 | 26.09 | 28.22 | 28.15 | 28.56 | 27.50 |
| 50 / 14.16 | 25.84 | 29.37 | 26.41 | 27.35 | 28.86 | 27.17 | 26.64 | 24.36 | 26.59 | 26.38 | 27.08 | 25.78 |
| 75 / 10.63 | 24.05 | 27.20 | 24.48 | 25.04 | 27.02 | 25.10 | 24.96 | 22.68 | 25.10 | 24.63 | 25.58 | 24.11 |
| 100 / 8.14 | 22.81 | 25.50 | 22.91 | 23.38 | 25.57 | 23.49 | 23.74 | 21.33 | 23.97 | 23.37 | 24.45 | 22.91 |